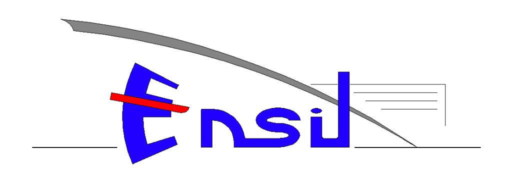


ENSIL-ELT – 2ème Année

TP 4- Ecran VGA

Travail préalable :

Dans ce TP le fonctionnement d'un écran VGA sera mis en évidence.

- Lisez les pages 15-17 du manuel d'utilisation de la carte Nexys 3.
- Télécharger de la page web du TP, le programme VHDL correspondant à l'écran VGA : VHDL → Carte SpartanIII de Digilent → programme correspondant à l'écran VGA.
- Analyser minutieusement ce programme. Expliquez la fonction des signaux X et Y.
- Faire correspondre le programme VHDL donné et le tableau de la page 17 du manuel.

ATTENTION : Ce programme a été fait initialement pour la carte Spartan III, pour une fréquence d'horloge de 50MHz, une petite modification est nécessaire pour l'adapter à la fréquence 100MHz de la carte Nexys 3.

Travail 1 : simulation

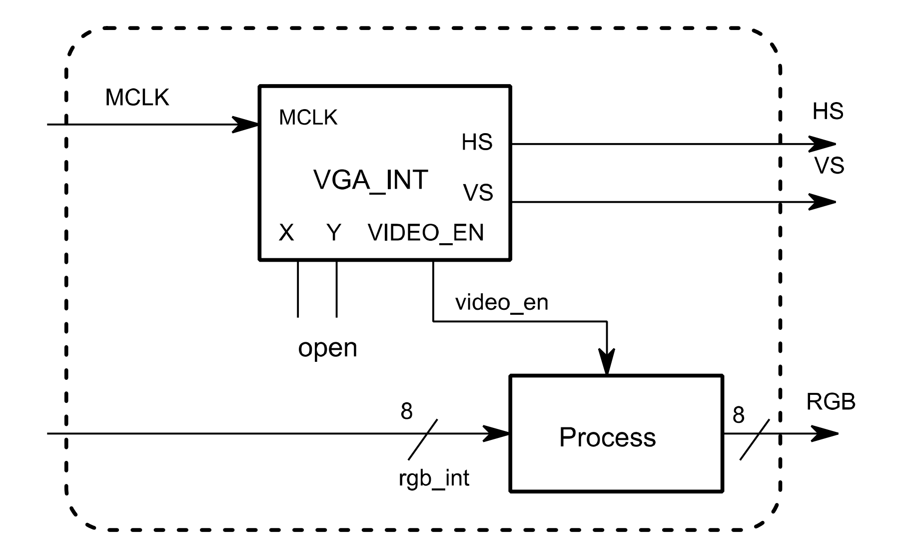
Nous allons effectuer une simulation pour vérifier le comportement de ce module.

- Créer un nouveau projet où vous ajouterez vga_int.vhd comme un module VHDL.
- Pour que la simulation ne soit pas trop longue, on propose de changer la taille de l'écran en 64x48 au lieu de 640x480. Effectuer les modifications nécessaires dans le programme vga_int.
- Créer un « test bench », lancer la simulation et vérifier les différents signaux de votre programme. Comparer les différents timings obtenus avec ceux du tableau du manuel (en tenant compte des modifications appliquées concernant un écran plus petit)

Travail 2 : écran multicolores

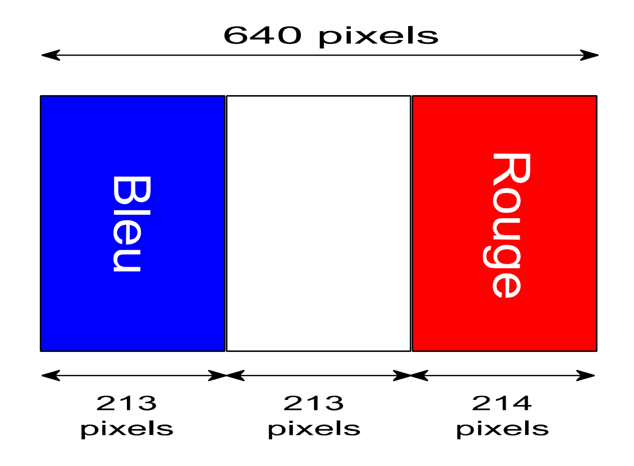
Quand vous êtes assuré du bon fonctionnement du programme, remettre le programme original (pour un écran 64x48). Nous allons faire afficher un écran jaune (rouge + vert).

- Faites de vga_int un component.
- Ecrire un programme principal « ecran.vhd » où vous ajouterez vga_int comme un component. Votre circuit doit afficher un écran couleur suivant l'état des interrupteurs de la carte. La figure ci-dessous présente l'architecture globale de votre programme principale.



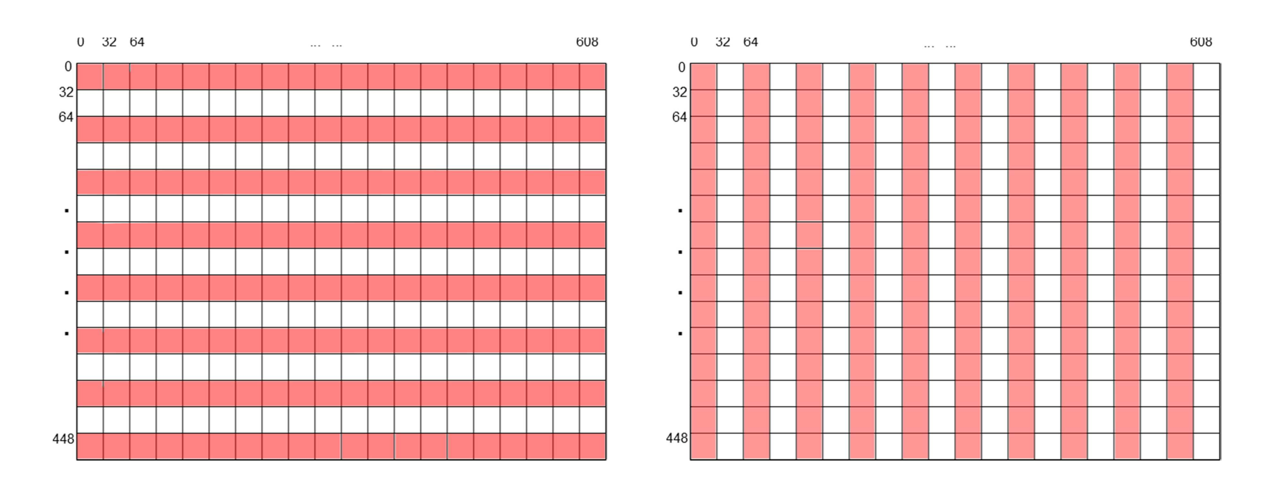
Travail 3

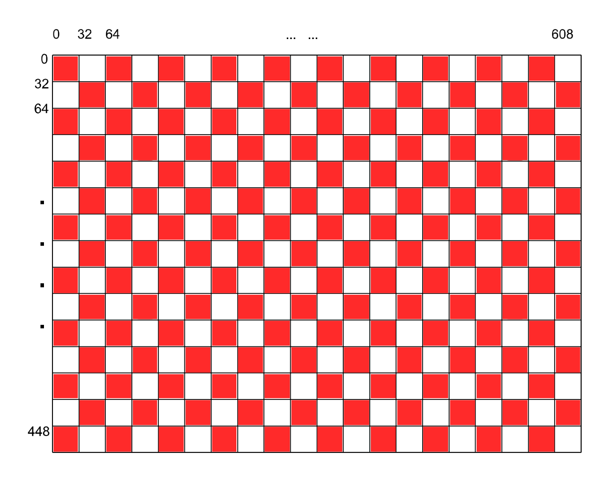
Faite afficher le drapeau français (dessiner d'abord le circuit avant de commencer à programmer).



Travail 4

- Créer les motifs suivant :





Travail 5

Si on tient compte de tous les bits de x et y, la résolution de l'écran VGA est de 640x480. Si on ignore le LSB et on ne considère que les 9 bits suivants (9 downto 1), la résolution de l'écran sera de 320x240. Et de la même manière :

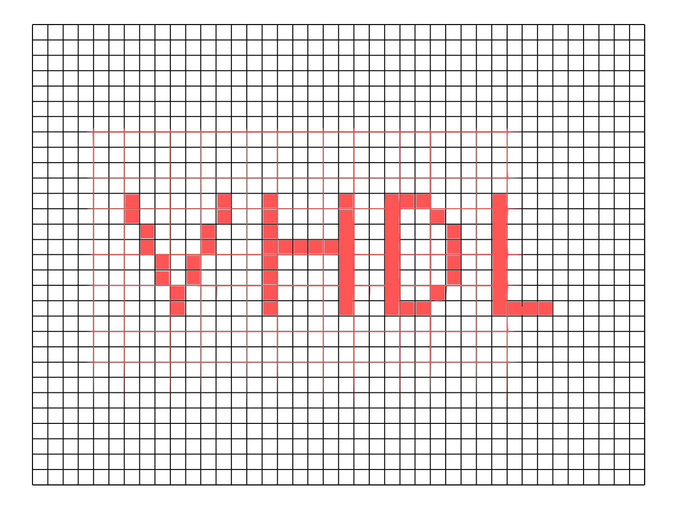
(9 downto 2) ➔ 160x120

(9 downto 3) ➔ 80x60

(9 downto 4) ➔ 40x30

On utilisera cette dernière résolution (40x30) pour afficher sur l'écran le mot VHDL.

l



Le motif (VHDL) est à sauvegarder dans une mémoire de taille 30 fois 40 bits de type ROM. Ci-dessous, un exemple de création ce type de mémoire :

```
entity rom is
    port (addr : in std_logic_vector(5 downto 0);
            data : out std_logic_vector (39 downto 0) );
end rom;

architecture fpga of rom is
constant data0: std_logic_vector(39 downto 0):=X"0010E42410"; -- 1ère ligne
constant data1: std_logic_vector(39 downto 0):=X"0011242410"; -- 2nde ligne
```

```
constant data2: std_logic_vector(39 downto 0):=X"0012242220"; -- 3ème ligne
constant data3: std_logic_vector(39 downto 0):=X"001227E220"; -- 4ème ligne
constant data4: std_logic_vector(39 downto 0):=X"0012242140"; -- 5ème ligne
constant data5: std_logic_vector(39 downto 0):=X"0012242140"; -- 6ème ligne
constant data6: std_logic_vector(39 downto 0):=X"0011242080"; -- 7ème ligne
constant data7: std_logic_vector(39 downto 0):=X"00F0E42080"; -- 8ème ligne
constant data8: std_logic_vector(39 downto 0):=X"0000000000"; -- ligne blanche

type rom_array is array (natural range 0 to 31) of std_logic_vector (39 downto 0);
constant rom_ecran : rom_array :=
(data8,data8,data8,data8,data8,data8,data8,data8,
data8,data8,data8,data0,data1,data2,data3,data4,
data5,data6,data7,data8,data8,data8,data8,data8,
data8,data8,data8,data8,data8,data8,data8, data8);
begin

    process(addr)
        variable j: integer range 0 to 31;
    begin
        j := conv_integer(addr);
        data <= rom_ecran(j);
    end process;
end fpga;
```

